Neurocomputing 143 (2014) 294–301

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Non-parallel support vector classifiers with different loss functions

Siamak Mehrkanoon*, Xiaolin Huang, Johan A.K. Suykens

KU Leuven, ESAT-STADIUS, Kasteelpark Arenberg 10, B-3001 Leuven (Heverlee), Belgium

ARTICLE INFO

Article history:
Received 3 April 2014
Received in revised form
11 May 2014
Accepted 17 May 2014
Communicated by Yukun Bao
Available online 12 June 2014

Keywords:
Non-parallel classifiers
Least squares loss
Pinball loss
Hinge loss
Kernel trick

ABSTRACT

This paper introduces a general framework of non-parallel support vector machines, which involves a regularization term, a scatter loss and a misclassification loss. When dealing with binary problems, the framework with proper losses covers some existing non-parallel classifiers, such as multisurface proximal support vector machine via generalized eigenvalues, twin support vector machines, and its least squares version. The possibility of incorporating different existing scatter and misclassification loss functions into the general framework is discussed. Moreover, in contrast with the mentioned methods, which applies kernel-generated surface, we directly apply the kernel trick in the dual and then obtain nonparametric models. Therefore, one does not need to formulate two different primal problems for the linear and nonlinear kernel respectively. In addition, experimental results are given to illustrate the performance of different loss functions.

1. Introduction

Support Vector Machines (SVM) is a powerful paradigm for solving pattern recognition problems [1,2]. In this method one maps the data into a high dimensional feature space and then constructs an optimal separating hyperplane in the feature space. This method attempts to reduce the generalization error by maximizing the margin. The problem is formulated as a convex quadratic programming problem. Least squares support vector machines (LSSVMs) on the other hand have been proposed in [3] for function estimation, classification, unsupervised learning, and other tasks [3,4]. In this case, the problem formulation involves equality instead of inequality constraints. Therefore in the dual one will deal with a system of linear equations instead of a quadratic optimization problem.

For binary classification problems, both SVMs and LSSVMs aim at constructing two parallel hyperplanes (or the hyperplanes in the feature space) to do classification. An extension is to consider non-parallel hyperplanes. The concept of applying two non-parallel hyperplanes was first introduced in [5], where two non-parallel hyperplanes were determined via solving two generalized eigenvalue problems and called GEPSVM. In this case one obtains two non-parallel hyperplanes where each one is as close as possible to the data points of one class and as far as possible from the data points of the other class. Recently many approaches, based on non-parallel hyperplanes, have been developed for classification, regression and feature selection tasks (see [6–11]).

The authors in [12] modified GEPSVM and proposed a non-parallel classifier called Twin Support Vector Machines (TWSVM) that obtains two non-parallel hyperplanes by solving a pair of quadratic programming problems. An improved TWSVM termed as TBSVM is given in [13] where the structural risk is minimized. Motivated by the ideas given in [3,14], recently least twin support vector machines (LSTSVM) are presented in [15], where the primal quadratic problems of TSVM are modified into least squares problem via replacing inequalities constraints by equalities.

In the above-mentioned approaches, kernel-generated surfaces are used for designing a nonlinear classifier. In addition one has to construct different primal problems depending on whether a linear or nonlinear kernel is applied. It is the purpose of this paper to formulate a non-parallel support vector machine classifier for which we can directly apply the kernel trick and thus it enjoys the primal and dual properties as in classical support vector machines classifiers. A general framework of non-parallel support vector machine, which consists of a regularization term, a scatter loss and a misclassification loss is provided. The framework is designed for multi-class problems. Several choices for the losses are investigated. The corresponding nonparametric models are given via considering the dual problems and the kernel trick.

The paper is organized as follows. In Section 2, a non-parallel support vector machine classifier with a general form is given. In Section 3, several choices of losses are discussed. The guidelines for the user are provided in Section 4. In Section 5, experimental results are given in order to confirm the validity and applicability of the proposed methods.

* Corresponding author. Tel.: +32 16328658; fax: +32 16321970.
E-mail addresses: siamak.mehrkanoon@esat.kuleuven.be (S. Mehrkanoon), xiaolin.huang@esat.kuleuven.be (X. Huang), johan.suykens@esat.kuleuven.be (J.A.K. Suykens).

http://dx.doi.org/10.1016/j.neucom.2014.05.063
0925-2312/© 2014 Elsevier B.V. All rights reserved.

2. Non-parallel support vector machine

Let us consider a given training dataset $\{x_i, y_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$, y_i is the label of the i-th data point and there are M number of classes. Here the one-vs-all strategy is utilized to build the codebook, i.e., the training points belonging to the m-th class are labeled by $+1$ and all the remaining data from the rest of the classes are considered to have negative labels. The index set corresponding to class m is denoted by $\mathcal{I}_m$. We seek non-parallel hyperplanes in the feature space:

$$f_m(x) = w_m^T \varphi_m(x) + b_m = 0, \quad m = 1, 2, \ldots, M$$

each of which is as close as possible to the points of its own class and as far as possible from the data points of the other class.

2.1. General formulation

In the primal, the hyperplane $f_m(x) = 0$ for class m can be constructed by the following problem:

$$\begin{aligned} \min_{w_m, b_m, e, \xi} \quad & \frac{1}{2} w_m^T w_m + \frac{\gamma_1}{2} \sum_{i \in \mathcal{I}_m} L_{(1)}(e_i) + \frac{\gamma_2}{2} \sum_{i \notin \mathcal{I}_m} L_{(2)}(\xi_i) \\ \text{subject to} \quad & w_m^T \varphi_m(x_i) + b_m = e_i, \quad \forall i \in \mathcal{I}_m \\ & 1 + (w_m^T \varphi_m(x_i) + b_m) = \xi_i, \quad \forall i \notin \mathcal{I}_m. \end{aligned} \tag{1}$$

After solving (1) for $m = 1, 2, \ldots, M$, we obtain M non-parallel hyperplanes in the feature space. Then the label of the new test point x^* is determined depending on the perpendicular distances of the test points from the hyperplanes. Mathematically, the decision rule can be written as follows:

$$\text{Label}(x^*) = \arg \min_{m=1,2,\ldots,M} \{d_m(x^*)\}, \tag{2}$$

where the perpendicular distance $d_m(x^*)$ is calculated by

$$d_m(x^*) = \frac{|w_m^T \varphi_m(x^*) + b_m|}{\|w_m\|_2}, \quad m = 1, 2, \ldots, M.$$

The target of (1) is to establish a hyperplane which is close to the points in class $\mathcal{I}_m$ and also is far away from the points that are not in this class. Therefore, any scatter loss function can be used for $L_{(1)}(\cdot)$ and at the same time any misclassification loss function can be utilized for $L_{(2)}(\cdot)$. Possible choices for $L_{(1)}(\cdot)$ include least squares, ϵ-insensitive tube, absolute, and Huber loss. For $L_{(2)}(\cdot)$, one can consider least squares, hinge, or squared hinge loss. Different loss has its own statistical properties and is suitable for different tasks. The proposed general formulation (1) is to handle multi-class problems, for which we essentially solve a series of binary problems. In the binary problem related to class m, we regard $x_i, i \in \mathcal{I}_m$ and the remaining points as two classes. Hence, the basic scheme of (1) for multi-class problems and binary problems is similar. For the convenience of expression, we focus on binary problems in theoretical discussion and evaluate multi-class problems in numerical experiments. Besides, for each class, one can apply different nonlinear feature mapping in (1). But in this paper, we discuss the case that unique $\varphi(x)$ is used for all the classes.

2.2. Related existing methods

For a binary problem, we assume that there are n_1 points in class 1 and n_2 points in class 2, i.e., there are n_1 elements in $\mathcal{I}_1$ and n_2 in $\mathcal{I}_2$. Suppose X_1 and X_2 are the matrices, of which each column is the vector $x_i, i \in \mathcal{I}_1$ and $x_i, i \in \mathcal{I}_2$, respectively. The corresponding matrices with feature mapping $\varphi(\cdot)$ are denoted by Φ_1 and Φ_2, i.e. the i-th row of Φ_1 is the vector $\varphi(x_i), i \in \mathcal{I}_1$, and so is Φ_2. Denote $Y_{n_1} = \text{diag}\{+1\}_{i=1}^{n_1} \in \mathbb{R}^{n_1 \times n_1}$, $Y_{n_2} = \text{diag}\{-1\}_{i=1}^{n_2} \in \mathbb{R}^{n_2 \times n_2}$, and 1_n as an n dimensional vector with all components equal to one. Then the non-parallel SVM (1) can be written in matrix formulation as the following two problems:

$$\begin{aligned} \min_{w_1, b_1, e, \xi} \quad & \frac{1}{2} w_1^T w_1 + \frac{\gamma_1}{2} L_{(1)}(e) + \frac{\gamma_2}{2} L_{(2)}(\xi) \\ \text{subject to} \quad & \Phi_1 w_1 + b_1 1_{n_1} = e \\ & Y_{n_2}[\Phi_2 w_1 + b_1 1_{n_2}] + \xi = 1_{n_2}, \end{aligned} \tag{3}$$

and

$$\begin{aligned} \min_{w_2, b_2, e, \xi} \quad & \frac{1}{2} w_2^T w_2 + \frac{\gamma_1}{2} L_{(1)}(e) + \frac{\gamma_2}{2} L_{(2)}(\xi) \\ \text{subject to} \quad & \Phi_2 w_2 + b_2 1_{n_2} = e \\ & Y_{n_1}[\Phi_1 w_2 + b_2 1_{n_1}] + \xi = 1_{n_1}. \end{aligned} \tag{4}$$

As discussed previously, $L_{(1)}(\cdot)$ could be any scatter loss function and any misclassification loss can be used in $L_{(2)}(\cdot)$. Some choices have been discussed. For example, if one chooses least squares loss for $L_{(1)}(\cdot)$ and hinge loss for $L_{(2)}(\cdot)$ and let $\gamma_1, \gamma_2 \to \infty$, the problem formulations (3) and (4), when a linear kernel is used, will reduce to TWSVM introduced in [12]:

$$\begin{aligned} \text{TWSVM1} \quad \min_{w_1, b_1, \xi} \quad & \frac{1}{2} \| X_1 w_1 + b_1 1_{n_1} \|^2 + C_1 1_{n_2}^T \xi \\ \text{subject to} \quad & -(X_2 w_1 + b_1 1_{n_2}) + \xi \geq 1_{n_2}, \end{aligned} \tag{5}$$

$$\begin{aligned} \text{TWSVM2} \quad \min_{w_2, b_2, \xi} \quad & \frac{1}{2} \| X_2 w_2 + b_2 1_{n_2} \|^2 + C_2 1_{n_1}^T \xi \\ \text{subject to} \quad & (X_1 w_2 + b_2 1_{n_1}) + \xi \geq 1_{n_1}. \end{aligned} \tag{6}$$

Another example is choosing least squares loss for both $L_{(1)}(\cdot)$ and $L_{(2)}(\cdot)$. Again, letting $\gamma_1, \gamma_2 \to \infty$ in (3) and (4) and using a linear kernel, one obtains the LSTSVM formulation reported in [15]

$$\begin{aligned} \text{LSTSVM1} \quad \min_{w_1, b_1, \xi} \quad & \frac{1}{2} \| X_1 w_1 + b_1 1_{n_1} \|^2 + \frac{C_1}{2} \xi^T \xi \\ \text{subject to} \quad & -(X_2 w_1 + b_1 1_{n_2}) + \xi = 1_{n_2}, \end{aligned} \tag{7}$$

$$\begin{aligned} \text{LSTSVM2} \quad \min_{w_2, b_2, \xi} \quad & \frac{1}{2} \| X_2 w_2 + b_2 1_{n_2} \|^2 + \frac{C_2}{2} \xi^T \xi \\ \text{subject to} \quad & (X_1 w_2 + b_2 1_{n_1}) + \xi = 1_{n_1}. \end{aligned} \tag{8}$$

In contrast with the classical support vector machines technique, TWSVM and LSTSVM do not take the structural risk minimization into account. For TWSVM, the authors in [13] gave an improvement by adding a regularization term in the objective function aiming at minimizing the structural risk by maximizing the margin. This method is called TBSVM, where the bias term is also penalized. But penalizing the bias term will not affect the result significantly and will change the optimization problem slightly. From a geometric point of view it is sufficient to penalize the norm of w in order to maximize the margin.

Another noticeable point is that TWSVM, LSTSVM, and TBSVM use a kernel generated surface to apply nonlinear kernels. As opposed to these methods, in our formulation, the burden of designing another two optimization formulations, when nonlinear kernel is used, is reduced by applying Mercer's theorem and kernel trick directly, which will be investigated in the following section.

3. Different loss functions

There are several possibilities for choosing the loss functions $L_{(1)}(\cdot)$ and $L_{(2)}(\cdot)$. Our target is to make the points in one class clustered in the hyperplane by minimizing $L_{(1)}(\cdot)$, which hence should be a scatter loss. For this aim, we prefer to use the least squares loss for $L_{(1)}(\cdot)$, because the related problem is easy to handle. Its weak point is that the least squares loss is sensitive to large outliers, then one may also consider ℓ_1-norm or Huber loss under the proposed framework. For $L_{(2)}(\cdot)$, which penalties misclassification error to push the points in other classes away from